ENHANCING IMAGE TRANSFER SECURITY IN FTP AND
SMTP PROTOCOLS

Jagadeeshwaran M\textsuperscript{1,*}, Jahnavi D\textsuperscript{2,*}, Sowjanya Lakshmi A\textsuperscript{3,*}

\textsuperscript{1,2,3}Student, ECE, Sastra Deemed University, Thanjavur, Tamil Nadu, India.

DOI: https://www.doi.org/10.56726/IRJMETS43980

ABSTRACT

When a user uploads their private data to the cloud, the data remains in the cloud as long as the service is running and even until the data is destroyed. In the given chain of events, there is no guarantee that the files will remain private and/or permanently deleted. A third party who is not the intended recipient of the file can view the file if he/she has access to the file storage. With most image-sharing services offered today, the image is transmitted over a protected channel but is not encrypted. This allows access to the files by an unintended recipient. In this project, we focus exclusively on image encryption. As images become an important part of digital communication, it is necessary to provide security and privacy during communication. To ensure security and user privacy, encryption is very important to protect the user from unauthorized access. Encryption is important because it allows you to securely protect data that you don't want anyone else to have access to. Encryption is a security method that encrypts information so that only authorized users can read it. It uses an encryption algorithm to generate cipher data that can only be read if decrypted.

Keywords: Cloud, Image sharing, Encryption, Cipher, Protocol, Decryption.

I. INTRODUCTION

The Exchange of files has been playing a major role in almost all sectors. There are various ways to exchange files which include physical methods like mail, transportation of files in hard copies, and also non-physical methods like the internet. The internet is playing a very important role in our lives. Now with the help of the internet, and with the usage of IP addresses, we are able to send and receive tons of information around the world without any physical connections and also in minimum time. Using FTP Socket programming and SMTP we can send files to any place around the globe while sitting in our homes. Privacy is the ability of an individual or group to confine themselves or data about themselves and thereby express themselves selectively. When something is private to a person, it is something that is inherently special or sensitive to them. The domain of privacy overlaps with security, which can incorporate the concepts of appropriate use, as well as the protection of information. We use simple XOR mathematical operations to ensure that security is achieved. Encryption is a way of scrambling data so that only authorized parties can understand the information. In technical terms, it is the process of converting human-readable plaintext to incomprehensible text, also known as ciphertext. In simpler terms, encryption takes readable data and alters it so that it appears random but not. Encryption requires the use of a cryptographic key: a set of mathematical values that both the sender and the recipient of an encrypted message have agreed upon.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{symmetric_encryption}
\caption{Symmetric Encryption}
\end{figure}

In this project, Symmetric encryption is used. It has a single key and uses the same key for both encryption and decryption, and must share the key with the entity that intends to communicate. AES and DES are some examples of symmetric encryption algorithms.
II. METHODOLOGY

OSI reference protocol architecture mode

Most network protocol suites are viewed as structured in layers. This is a result of the Open Systems Interconnect (OSI) Reference Model designed by the International Standards Organization (ISO). The OSI model describes network activities as having a structure of seven layers, each of which has one or more protocols associated with it. The layers represent data transfer operations common to all types of data transfers among cooperating networks. The protocol layers of the OSI Reference Model are traditionally listed from the top (layer 7) to the bottom (layer 1) up, as shown in the following table. The operations defined by the OSI model are conceptual and not unique to any particular network protocol suite. For example, the OSI network protocol suite implements all seven layers of the OSI Reference Model. FTP and SMTP use some of the OSI model layers.

<table>
<thead>
<tr>
<th>Layer No.</th>
<th>Layer Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Application</td>
<td>Consists of standard communication services and applications that everyone can use</td>
</tr>
<tr>
<td>6</td>
<td>Presentation</td>
<td>Ensures that information is delivered to the receiving machine in a form that it can understand</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td>Manages the connections and terminations between cooperating computers</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td>Manages the transfer of data and assures that received and transmitted data are identical</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
<td>Manages data addressing and delivery between networks</td>
</tr>
<tr>
<td>2</td>
<td>Data Link</td>
<td>Handles the transfer of data across the network media</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
<td>Defines the characteristics of the network hardware</td>
</tr>
</tbody>
</table>

Figure 2: Open Systems Interconnection Reference Mode

Sockets and socket programming

It is a way of connecting two or more nodes in a network for communication. A socket is an endpoint between two-way communication of a network. One socket listens or receives data on a particular port at an IP, while the other sockets reach out to others to form a stable connection. A server forms the listener socket while the client reaches out to the server. Many protocols like FTP, SMTP, TCP, and POP3 make use of sockets to establish a connection between client and server for the exchange of data.

Figure 3: The Socket Interface

Figure 3 gives us an idea about the overlay of sockets and how they are used to connect two devices with different IP addresses to connect wirelessly through the internet. In this way, we can share data, send information, and receive information through the internet using a socket by proper design of a socket program.
File Transfer Protocol (FTP)

The usage of this protocol is to transfer files between an FTP host(server) and an FTP client computer on the internet. It is also used to download files from the World Wide Web. It is a replacement for HTTP for downloading and uploading files to FTP servers. FTP remains one of the fundamental building blocks of computer networking and the modern internet. It was developed in the early 1970s initially to allow the secure transfer of files between the server and the client over the ARPANET Network Control Program. The initial development was simple and did not support well for security concerns. Today, with security as the major concern, FTP is being supported by SFTP and FTPS to ensure secure(encrypted) and reliable transfer of files between two computers (remote and local computers). FTP uses one connection for commands and the other for sending and receiving data. A logical connection point for communication using the Internet Protocol is called a port. Port 21 is used for sending commands and is the standard port (also known as the command port). The data port is used for transferring files. The data port’s number will depend upon the type of mode used.

Figure 4: Working of FTP

There are two types of transfer modes available in FTP. The type of file we send decides on which mode an FTP service works. American Standard for Information Interchange (ASCII) mode deals with any text files (HTML). The structure of binary mode is entirely different. The default mode for the transfer setting is the binary mode. This is because sending both file types in binary mode is easier. ASCII mode suits well for text-based file transfers. Texts are transferred much faster in this mode. Files to be transferred in ASCII mode: Text files, HTML files, and CGI scripts. Files to be transferred in Binary mode: Any files that do not contain pure text. Images, applications, .zip files. FTP relies on two communication channels between the server and the client.

Fig. 2.3.2 FTP Channels

Figure 5: FTP Channels

Control connection: The FTP client sends a connection request to port number 21. This control connection is used for sending and receiving responses.

Data connection: This connection is for transferring files and folders. This connection is achieved in two ways. They are active and passive modes.

Active mode: In this mode, the client opens a port and listens and the server actively connects to it. The user connects from a random port on the client to port 21 of the server. It sends a PORT command which tells the server what port of the client it should connect to. The server connects to port 20, the port that the client had designated. Data transfer happens when the connection is established.

Passive mode: Here, the server opens a port and listens passively and the client connects to it. This is the default connection mode which offers more security. Firewalls usually block connections that come from the
outside. In this mode, the client reaches out to the server to make the connection. The client connects from the random port to the port that the server has designated.

Usually, a user needs to log in to the server to establish a connection but there are some servers whose content can be shared without login credentials. These are called anonymous FTP. We use anonymous FTP in this project. Even though FTP is simple and faster than HTTP, its usage led to potential threats to users. The first and foremost concern is security. Transparency of the data and command connections happened to be a serious problem and they made it easy for the third-party attacker/interloper to access the user's data and login credentials. Keeping this in mind, we developed this project which implements FTP but with additional security.

**Sockets and socket programming Simple Mail Transfer Protocol (SMTP)**

In ancient times, communication between large distances was very difficult. In our digitized era communication becomes very handy with the use of technology. The E-mail concept was introduced by Ray Tomlinson in 1971.

The E-mail concept is the best way to data transfer. SMTP was originally authored by Jonathan Postel in 1982. SMTP is the network protocol used for data transfer at the receiving end. MTA is the server process running on a mail exchange host that is responsible for routing and sending mail. It can also send a single message to multiple users. The message can be of image, text, audio, or video type. Login credentials are optional in SMTP and only required if the server has SMTP authentication. SMTP is part of the Application Layer of the TCP/IP protocol. It instructs how the mail must move from one MTA to another MTA. Mails might move to many mail servers before reaching the recipient's server.

SMTP uses the "Store and Forward" feature to send emails. SMTP provides a set of codes that simplify the communication of email between servers. It breaks the mail into chunks for easy transmission. SMTP works on port 25 (standard for mail transfer), port 465 (not compliant with RFC), and port 587 (TLS encrypted). The domain name service or the server address for g-mail is 'smtp.gmail.com' and the port number is 587. The domain name service or the server address for yahoo is 'smtp.mail.yahoo.com' and the port number is 465.

**Methodology**

1. Connect to the server
2. Login with the required credentials
3. Make service request
4. Quit the server

**Working**

Initially, we have to connect to the server. The mail is composed and the mail is carried by the MTA agent to the server of the client with the required credentials. `login()` method is used to get the credentials. `starttls()` method is used for the server authentication. We have to make a service request by giving the correct domain name service and port number for g-mail and yahoo services. `quit()` method is used to terminate the connection between the servers. If the address (domain name service) of the sender and receiver is different, MSA will
force the MTA to quit or terminate the connection abruptly. MIME allows the transfer of non-ASCII characters. Once the message is received at the recipient's address, MDA will store the mail at the receiving end. IMAP4 and POP3 are the most widely used protocol used along with the SMTP to retrieve the message for the hosts.

Advantages
➔ SMTP provides the simplest form of email communication
➔ Emails can be sent quickly and effectively
➔ Offers reliability in terms of outgoing mail

Disadvantages
➔ SMTP can only send plain text messages, not fancy items like attachments or fonts. This part is done by the MIME. It converts non-text elements to plain text
➔ SMTP is limited to 7-bit ASCII characters
➔ Some firewalls may block common ports required by SMTP

III. IMPLEMENTATION

Python programming language
Python is a dynamic programming language used by developers all over the world in the present era. Having its origin in 1994 (the release of Python 1.0), Python today has multiple implementations and applications. Some of the applications are,
1) GUI-based applications
2) Games
3) Scientific applications
4) Cryptography
5) Computational applications
6) Web frameworks and applications
7) Operating systems
8) Network programming
9) Data Science and Machine Learning

Now using Python language, we can produce a program to create a network connection between two computers Using FTP and share data, or even create new data on the remote computer. Knowing how to do this, makes us able to share data without using third-party apps and so increasing privacy and security. We are using this for sending complex data like encrypted data, or confidential files. We can also send the encrypted data to the client via email using SMTP which is again implemented using Python programming language. In this project, we will encrypt/decrypt an image using simple mathematical logic. It requires two things, data, and key, and when XOR operation is applied on both the operands i.e., data and key, the data gets encrypted but when the same process is done again with the same key-value data gets decrypted.

Encryption
It is nothing but a simple process in which we convert our data or information into secret code to prevent it from unauthorized access and keep it private and secure. Encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot. Encryption does not itself prevent interference but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm – a cipher – generating ciphertext that can be read only if decrypted. Early encryption techniques were often utilized in military messaging. Since then, new techniques have emerged and become commonplace in all areas of modern computing.

Decryption
The conversion of encrypted data into its original form is called Decryption. It is generally a reverse process of encryption. It decodes the encrypted information so that an authorized user can only decrypt the data because requires a secret key or password. As information travels over the Internet, it is necessary to scrutinize access
from unauthorized organizations or individuals. Due to this, the data is encrypted to reduce data loss and theft. A few common items that are encrypted include text files, images, e-mail messages, user data, and directories. Decryption can be done manually or automatically. It may also be performed with a set of keys or passwords. In this project, here we will again apply the same XOR operation on an encrypted image to decrypt it. But always remember that our encryption key and decryption key must be the same.

**Demonstration**

Imagine Jane has an important message that wants to send to John (over a network). So Jane generates a symmetric key (K1) and shares it with John prior to the communication. Both of them will come to an agreement that they will use the K1 key for both encryption and decryption of the messages they pass between them. So Jane encrypts the message and sends it to John over the network. John will decrypt the message on the receiving end and will get the original message. Since only Jane and John have the key, others cannot read the message they share even though they have access to it.

**OUTPUT SNAPSHOTS**

**Secure image transfer using SMTP**

1) While executing, the console is asking for what kind of work it has to do.

```plaintext
A "EDITNEWS.3.11"
File Edit File Debug Options Window Help

Python 3.9.11 (tags/v3.9.11:30e6c74, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>> RESTART: D:\Project CW\Secure Image Transfer.py
Greetings!!

Press 1 if you want to send image securely
Press 2 if you want to decrypt the image sending by server
Press 3 for encrypting the image received by your email
Enter your response: 1
Press 1 for LAN transfer or Press 2 for Small transfer: 2
Encryption / Decryption starts...

Figure 8
```

2) We are giving ‘1’ as a response because here we are encrypting and ‘2’ as another input as we are sending an image to the e-mail client.
3) Now we have to give any number between 1 to 255. It will be used as a key and now press the button 'encrypt/decrypt'.

4) We can choose any .jpg image from any folder. After choosing click the ‘Open’ button and close the ‘tk’ file box.
5) Image is successfully encrypted.

6) Now, we have to give the client's email addresses one by one. We can send to any number of clients at one go.

7) Emails are sent successfully. The server has connected and closed each time for each and every client.
8) Both of the clients received the image. Now the file is encrypted so no one can view the image.

9) Clients can download from email and save the image wherever they want. As it is encrypted no one can view the file.
10) Clients should run the code and give '3' as a response for decrypting the image. They should give the same key used by the server to encrypt the image.

![Figure 18](image18.png)

11) An open file box will appear in which the clients should select the encrypted image. Now the decryption is successfully done and the image is saved where this python executable file is saved in the client's computer.

![Figure 19](image19.png)

12) Image is successfully decrypted and now the client can view the image.

**Secure image transfer using FTP**

1. We are giving '1' as a response to encrypt any image and giving '1' as a second input as here we using LAN to transfer the images.

![Figure 20](image20.png)
2. After giving any number between 1 to 255 to be used as a key and now press the button 'encrypt/decrypt'. We can choose any .jpg image from any folder. After choosing click the 'Open' button and close the 'tk' file box.

![Figure 21](image1.png)

3. Now, we can give any valid port number. IP will be detected and printed automatically. Now the server is ready to send the image.

![Figure 22](image2.png)

4. Client and server are connected via LAN in this case we connected the client to the server's hotspot. The client should run the executable file and give '2' as the response. Now the client should enter the port number which we gave on the server side. Once done the encrypted image is successfully downloaded.

![Figure 23](image3.png)
5. Now the client should give the same key used by the server to encrypt the image.

![Figure 24](image1.png)

6. In the open file box select the encrypted image. No one can view the image at this point of time. After selecting click the 'Open' button and close the 'tk' file box.

![Figure 25](image2.png)

7. The image is now successfully decrypted.

![Figure 26](image3.png)
8. The decrypted image is saved where this python executable file is saved in the client’s computer.

![Image of a computer screen with a file management interface]

Figure 27

V. CONCLUSION

We have realized the fact that privacy is a fundamental right of an individual and it is absolutely essential. We have come up with this idea for the project after researching the fact that modern File Storage and Sharing service providers receive the file unencrypted from the user. This means that the file data is easily available to the service provider or anyone with access to the database.

Our project protects the user’s data from the service provider and any other third parties from accessing it. We believe that our project will be celebrated by those who value privacy the most and it will be part of their day-to-day lives while sending data and receiving data to their family, friends, and colleagues.

Instead of relying on a third-party application and resources available on the internet, a group working on the same LAN or internet can use this code to share image files securely. The main reason to use the code is that this is not a public platform so many will not know about our implementation and our details.

We intend to improve this project by adding more features, enhancing its security, and releasing it to the general public. We also plan to develop an Android Application of the same to give our users even more flexibility instead of restricting file transfers to just the computer. We try to include all types of images not only the .jpg type. Run time exceptions will be handled in the future.

All the main points of the research work are written in this section. Ensure that abstract and conclusion should not same. Graph and tables should not use in conclusion.

VI. REFERENCES


